eSi-DMA

1 Contents

2 Overview

The eSi-DMA core can be used to implement memory-to-memory, memory-to-peripheral, peripheral-to-memory and peripheral-to-peripheral block data transfers. It supports the following features:

- Configurable number of channels.
- Configurable number of peripherals (up to 64).
- Programmable byte count, access size and burst length.
- Programmable addressing (incrementing / fixed).
- Fixed priority (lowest channel has highest priority).
- AMBA 3 AHB-lite slave interface for control register access.
- AMBA 3 AHB-lite master interface for data transfers.

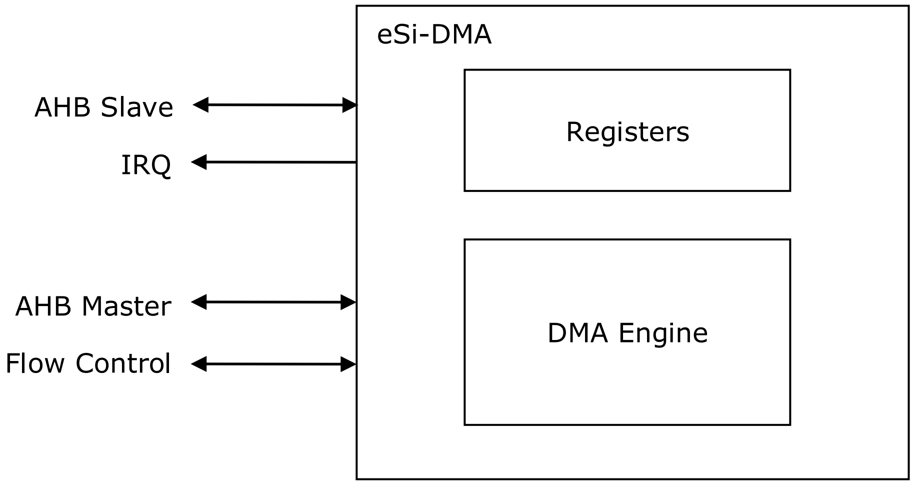

Figure 1: eSi-DMA

3 Hardware Interface

Module Name	cpu_ahb_dma
HDL	Verilog
Technology	Generic
Source Files	cpu_ahb_dma.v, cpu_fifo.v

Port	Type	Description
channels	Integer	Specifies the number of channels implemented
peripherals	Integer	Specifies the number of peripherals connected
fifo_depth	Integer	Specifies the depth of the FIFO used for holding data during transfers

Table 1: Parameters

Port	Direction	Width	Description
s_hclk	Input	1	Slave interface, AHB clock
s_hresetn	Input	1	Slave interface, AHB reset, active-low
s_haddr	Input	BITS	Slave interface, AHB address
s_hburst	Input	3	Slave interface, AHB burst type
s_hmastlock	Input	1	Slave interface, AHB locked transfer
s_hprot	Input	4	Slave interface, AHB protection
s_hsize	Input	3	Slave interface, AHB size
s_htrans	Input	2	Slave interface, AHB transfer type
s_hwdata	Input	BITS	Slave interface, AHB write data
s_hwrite	Input	1	Slave interface, AHB write
s_hready	Input	1	Slave interface, AHB ready
s_hsel	Input	1	Slave interface, AHB select
s_hready	Output	1	Slave interface, AHB ready
s_hrdata	Output	BITS	Slave interface, AHB read data
s_hresp	Output	1	Slave interface, AHB response
m_hclk	Input	1	Master interface, AHB clock. Must be the same frequency and synchronous to s_hclk
m_hresetn	Input	1	Master interface, AHB reset, active-low
m_hready	Input	1	Master interface, AHB ready
m_hrdata	Input	BITS	Master interface, AHB read data
m_hresp	Input	1	Master interface, AHB response
m_haddr	Output	BITS	Master interface, AHB address
m_hburst	Output	3	Master interface, AHB burst type
m_hmastlock	Output	1	Master interface, AHB locked transfer
m_hprot	Output	4	Master interface, AHB protection
m_hsize	Output	3	Master interface, AHB size
m_htrans	Output	2	Master interface, AHB transfer type
m_hwdata	Output	BITS	Master interface, AHB write data
m_hwrite	Output	1	Master interface, AHB write
tx_ready	Input	peripherals	Indicates peripheral can accept new data
rx_ready	Input	peripherals	Indicates peripheral has data to be read
tx_ack	Output	peripherals	Acknowledges tx_ready after transfer complete
rx_ack	Output	peripherals	Acknowledges rx_ready after transfer complete

Table 2: I/O Ports

For complete details of the AHB signals, please refer to the AMBA 3 AHB-Lite Protocol v1.0 Specification available at http://www.arm.com/products/solutions/AMBAHomePage.html

The DMA does not include internal synchronizing flip-flops. These should be implemented externally for the `rx_ready` and `tx_ready` ports if the transmitting clock domain is asynchronous to `m_hclk`.

4 Software Interface

4.1 Register Map

Each DMA channel has its own set of registers, as illustrated in Table 3: Register Map. In this table, `N`, indicates the channel number, which ranges from 0 to `channels`-1.

Register	Address offset	Access	Description
`src_address[N]`	0x20*N+0x00	R/W	Source address register
`dst_address[N]`	0x20*N+0x04	R/W	Destination address register
`src_control[N]`	0x20*N+0x08	R/W	Source control register
`dst_control[N]`	0x20*N+0x0c	R/W	Destination control register
`count[N]`	0x20*N+0x10	R/W	Count register
`status[N]`	0x20*N+0x14	R/W	Status register
`control[N]`	0x20*N+0x18	R/W	Control register

Table 3: Register Map

4.1.1 Source Address Register

The source address register contains the base address of the memory block to be copied. The source address must be aligned according to the value of `SIZE` in the `src_control` register.

BITS-1 … 0

Figure 2: Format of the `src_address` register

4.1.2 Destination Address Register

The destination address register contains the base address of the location to copy the memory block to. The destination address must be aligned according to the value of `SIZE` in the `dst_control` register.

BITS-1 … 0

Figure 3: Format of the `dst_address` register

4.1.3 Source Control Register

The source control register contains a selection of flags that control the operation of the DMA channel with respect to the source data.

15	14	13	12	11	10	9	8	7		5	4		2	1	0
PIDX						-		BURST			SIZE			T	A

Figure 4: Format of the `src_control` register

Register	Values	Description
`A`	0 – Increment address 1 – Fixed address	How to update the source address after each access
`T`	0 – Memory 1 – Peripheral	Source address type. When set to peripheral type, the `rx_ready[PIDX]` signal from the

		peripheral must be asserted before a transfer will take place. When set to memory, the transfer will occur unconditionally
`SIZE`	0 – 1 byte 1 – 2 bytes 2 – 4 bytes (`BITS` >= 32)	Size of each access. This must be identical to the destination `SIZE` setting
`BURST`	0 – 1 beat 1 – 4 beats 2 – 8 beats 3 – 16 beats	Length of burst. This must be identical to the destination `BURST` setting
`PIDX`	0 – `peripherals`-1	Peripheral index. Determines which of the `rx_ready` signals should be used for flow control when source address type is set to peripheral

Table 4: Fields of the `src_control` register

4.1.4 Destination Control Register

The destination control register contains a selection of flags that control the operation of the DMA channel with respect to the destination data.

15–10	9–8	7–5	4–2	1	0
PIDX	-	BURST	SIZE	T	A

Figure 5: Format of the `dst_control` register

Register	Values	Description
`A`	0 – Increment address 1 – Fixed address	How to update the destination address after each access
`T`	0 – Memory 1 – Peripheral	Destination address type. When set to peripheral type, the `tx_ready[PIDX]` signal from the peripheral must be asserted before a transfer will take place. When set to memory, the transfer will occur unconditionally
`SIZE`	0 – 1 byte 1 – 2 bytes 2 – 4 bytes (`BITS` >= 32)	Size of each access. This must be indentical to the source `SIZE` setting
`BURST`	0 – 1 beat 1 – 4 beats 2 – 8 beats 3 – 16 beats	Length of burst. This must be identical to the source `BURST` setting
`PIDX`	0 – `peripherals`-1	Peripheral index. Determines which of the `rx_ready` signals should be used for flow control when source address type is set to peripheral

Table 5: Fields of the `dst_control` register

4.1.5 Count Register

The count register contains the total number of bytes that should transfered. For transfers with `SIZE`=2, count must be divisible by two. Similarly, for transfers with `SIZE`=4, count must be divisible by four.

BITS-1	0

Figure 6: Format of the `count` register

4.1.6 Status Register

The status register contains a selection of flags that indicate the current status of the DMA channel. The `TC` flag is read-only. To clear the `ER` flag, write a 1 to it. Writing 0 will leave it unchanged.

	1	0
-	ER	TC

Figure 7: Format of the `status` register

Register	Values	Description
`TC`	0 - Not complete 1 – Complete	Transfer complete
`ER`	0 – No error 1 – Error	Indicates an error occurred during the transfer

Table 6: Fields of the `status` register

4.1.7 Control Register

The - register contains a selection of flags that control the operation of the DMA channel.

	3	2	1	0
-	SP	ERIE	TCIE	E

Figure 8: Format of the `control` register

Register	Values	Description
`E`	0 - Disabled 1 - Enabled	Enables the DMA channel
`TCIE`	0 – Disabled 1 – Enabled	Transfer complete interrupt enable
`ERIE`	0 – Disabled 1 – Enabled	Error interrupt enable
`SP`	0 – Continue 1 – Stop	Stop transfer immediately. Data may be lost

Table 7: Fields of the `control` register

4.2 Interrupts

The DMA supports the following interrupts.

- Per-channel transfer complete interrupt
- Per-channel error interrupt

The transfer complete interrupt will be raised when the DMA has transfered the number of bytes specified in the `count` register. The `TC` flag in the `status` register will be set 1 to indicate this. When the `TX` flag in the `status` register is set to 1 and the the `TCIE` flag in the `control` register is set to 1, the transfer complete interrupt will be asserted.

The error interrupt will be raised then the `ER` flag in the `status` register is 1 and the `ERIE` flag in the `control` register is set to 1. This indicates an error was detected during the transfer.

5 Revision History

Hardware Revision	Software Release	Description
1	2.4.0	Initial release

Table 8: Revision History